



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ФУНДАМЕНТАЛЬНЫЕ НАУКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И МАТЕМАТИЧЕСКАЯ
ФИЗИКА (ФН11)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 02.03.01 МАТЕМАТИКА И КОМПЬЮТЕРНЫЕ
НАУКИ

О Т Ч Е Т

по домашнему заданию № 2

Вариант №20

Название: *Часть 2.1. Обработка одномерных массивов.*
Часть 2.2 Обработка матриц.
Часть 2.3. Декомпозиция и разработка подпрограмм.

Дисциплина: Информатика

Студент

ФН11-12Б
(Группа)

ХФ 17.11.2020
(Подпись, дата)

М.Х.Хаписов
(И.О. Фамилия)

Преподаватель


(Подпись, дата)

Т.Н.Ничушкина
(И.О. Фамилия)

17.11.2020

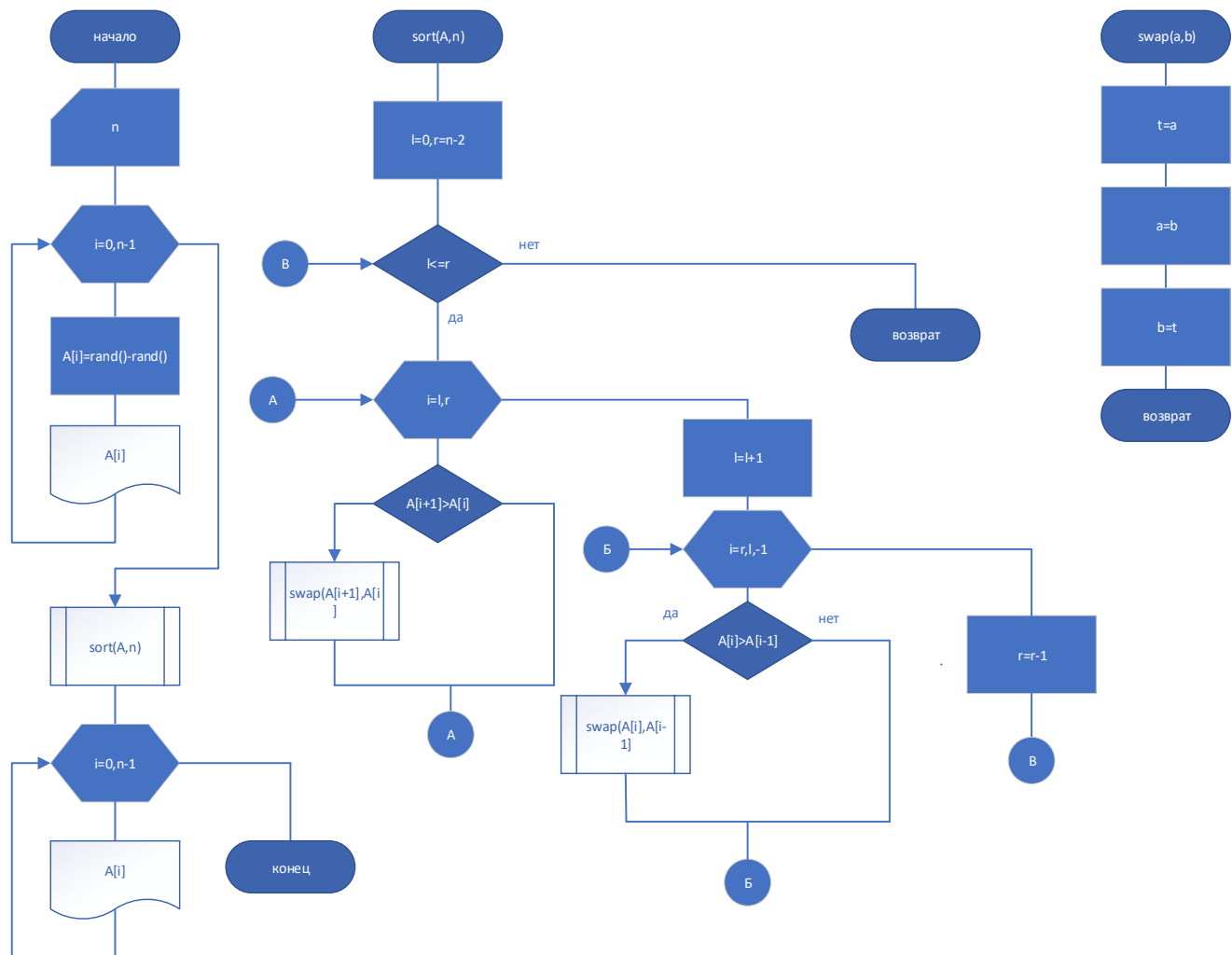
Москва, 2020

Обработка одномерных массивов

Цель: научиться обрабатывать одномерные массивы

Задача: упорядочить одномерный массив по невозрастанию значений его элементов, используя метод Шейкерной сортировки

Схема алгоритма



Текст программы

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void swap(int &a,int &b) {
    int t=a;
    a=b;
    b=t;
}
void sort(int *A,int n) {
    int l=0,r=n-2;
    while (l<=r) {
        for (int i=l;i<=r;i++) if (A[i+1]>A[i]) swap(A[i+1],A[i]);
        l++;
        for (int i=r;i>=l;i--) if (A[i]>A[i-1]) swap(A[i],A[i-1]);
        r--;
    }
}
void main() {
    srand(time(NULL));
    int n;
    puts("Input a number of elements in the array");
    scanf_s("%d",&n);
    while ((n<1)|| (n>50)) {
        puts("A number of elements in the array must be integer number,
which is more than zero and less than 51");
        scanf_s("%d",&n);
    }
    int *A=new int[n];
    puts("Input array:");
    for (int i=0;i<n;i++) { A[i]=rand()-rand(); printf("%d ",A[i]); }
    sort(A,n);
    printf("\n");
    puts("Output array:");
    for (int i=0;i<n;i++) printf("%d ",A[i]);
    printf("\n");
    system("pause");
}
```

Тестирование

```
C:\Users\Knigan\source\repos\Project3\Debug\Project3.exe
Input a number of elements in the array
16
Input array:
4133 5850 4315 4034 15227 -11933 96 -22145 -13868 -2263 9757 -23201 5776 -3441 5021 5423
Output array:
15227 9757 5850 5776 5423 5021 4315 4133 4034 96 -2263 -3441 -11933 -13868 -22145 -23201
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project3\Debug\Project3.exe
Input a number of elements in the array
15
Input array:
-21410 25074 -6990 -962 -4285 3185 -258 -5279 13795 -7776 -5221 8847 258 15464 -12102
Output array:
25074 15464 13795 8847 3185 258 -258 -962 -4285 -5221 -5279 -6990 -7776 -12102 -21410
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project3\Debug\Project3.exe
Input a number of elements in the array
14
Input array:
-17150 7367 -17521 12919 15259 -1422 2891 -19853 -14154 28625 -4487 4243 19783 381
Output array:
28625 19783 15259 12919 7367 4243 2891 381 -1422 -4487 -14154 -17150 -17521 -19853
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project3\Debug\Project3.exe
Input a number of elements in the array
13
Input array:
-13954 -5914 -844 -9438 -2852 -4877 5252 1984 -26924 -1418 -12128 -15592 18042
Output array:
18042 5252 1984 -844 -1418 -2852 -4877 -5914 -9438 -12128 -13954 -15592 -26924
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project3\Debug\Project3.exe
Input a number of elements in the array
12
Input array:
-1079 -8568 3885 -8537 -15739 -6989 1234 -10158 -140 -10675 8688 6874
Output array:
8688 6874 3885 1234 -140 -1079 -6989 -8537 -8568 -10158 -10675 -15739
Для продолжения нажмите любую клавишу . . .
```

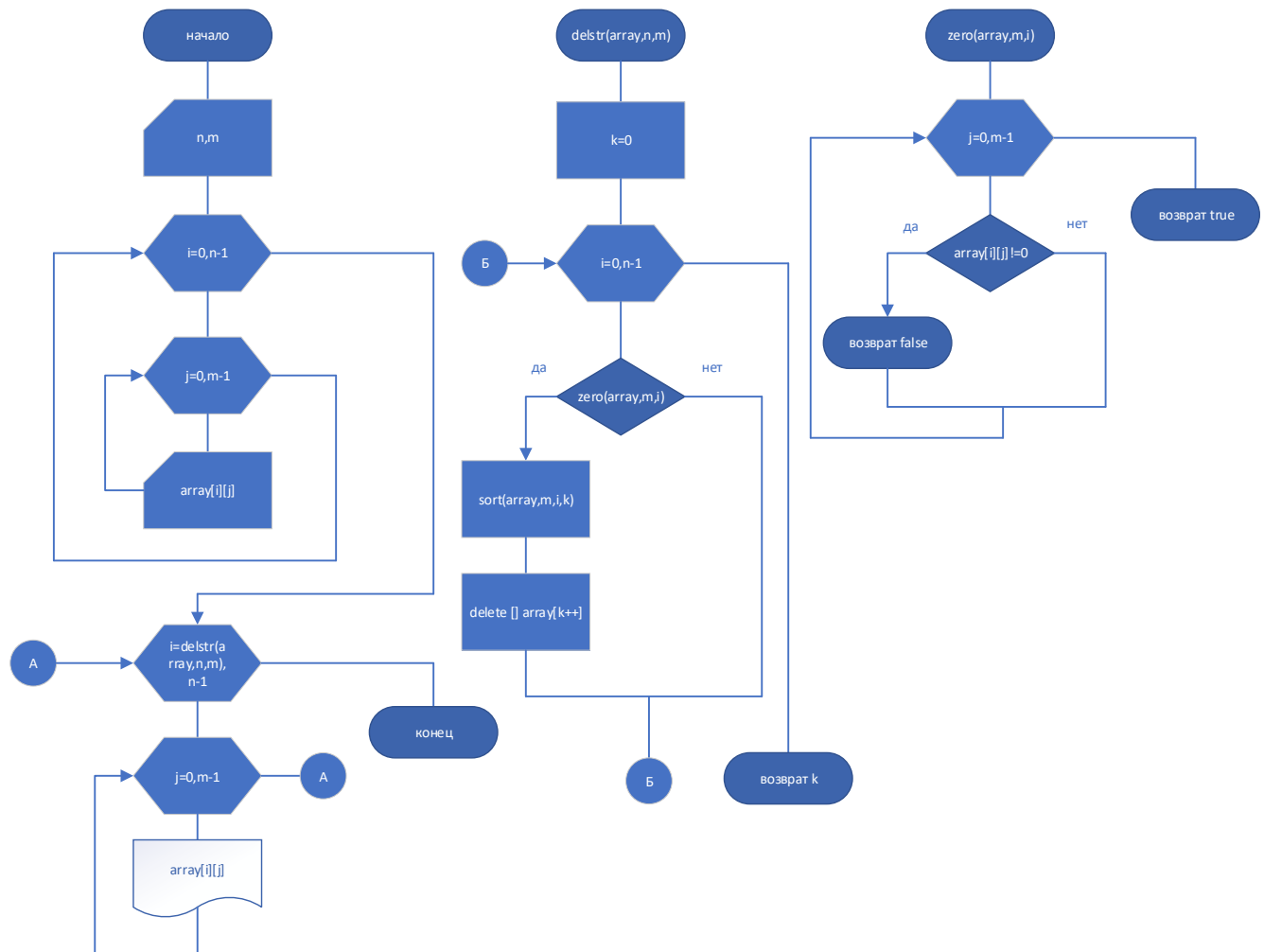
Вывод: я научился обрабатывать одномерные массивы и освоил метод Шейкерной сортировки

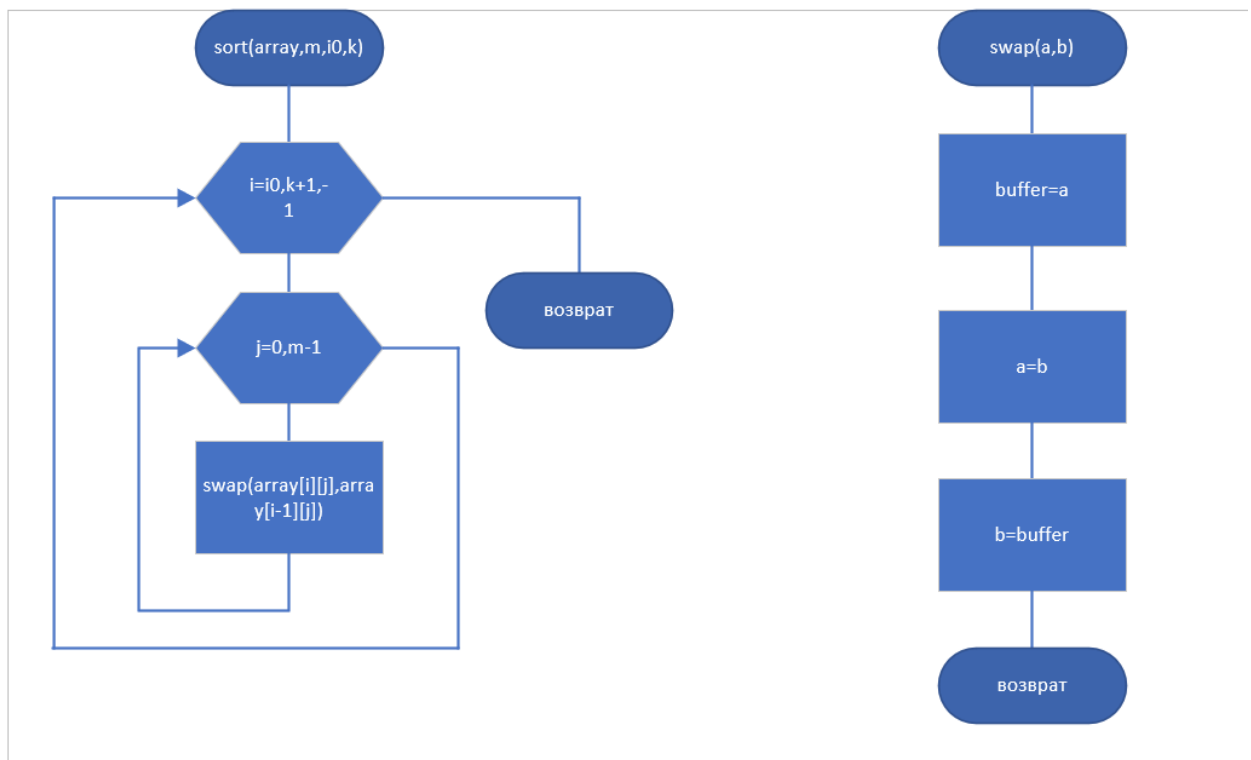
Обработка матриц

Цель: научиться обрабатывать матрицы

Задача: удалить все строки матрицы, содержащие нули

Схема алгоритма





Текст программы

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
bool zero(int **array,int m,int i) {
    for (int j=0;j<m;j++)
        if (array[i][j]!=0) return false;
    return true;
}
void swap(int &a,int &b) {
    int buffer=a;
    a=b;
    b=buffer;
}
void sort(int **array,int m,int i0,int k) {
    for(int i=i0;i>k;i--)
        for(int j=0;j<m;j++) swap(array[i][j],array[i-1][j]);
}
int delstr(int **array,int n,int m) {
    int k=0;
    for (int i=0;i<n;i++) {
        if (zero(array,m,i)) {
            sort(array,m,i,k);
            delete [] array[k++];
        }
    }
    return k;
}
void main() {
    int **array,n,m;
    puts("Enter array's size (n,m)");
    scanf_s("%d %d",&n,&m);
    while ((n>10)|| (m>10)|| (n<1)|| (m<1)) {
        puts("Error! 0<n,m<11");
        scanf_s("%d %d",&n,&m);
    }
}

```

```

array=new int*[n];
for (int i=0;i<n;i++) array[i]=new int[m];
puts("Enter array");
for (int i=0;i<n;i++)
    for (int j=0;j<m;j++) scanf_s("%d",&array[i][j]);
puts("Input array:");
for (int i=0;i<n;i++) {
    for (int j=0;j<m;j++) printf("%d ",array[i][j]);
    printf("\n");
}
puts("Output array:");
for (int i=delstr(array,n,m);i<n;i++) {
    for (int j=0;j<m;j++) printf("%d ",array[i][j]);
    printf("\n");
}
system("pause");
}

```

Тестирование

```

C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Enter array's size (n,m)
3 3
Enter array
0 0 0
1 1 1
2 2 2
Input array:
0 0 0
1 1 1
2 2 2
Output array:
1 1 1
2 2 2
Для продолжения нажмите любую клавишу . . .

```

```

C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Enter array's size (n,m)
4 3
Enter array
0 0 0
1 1 1
2 2 2
0 0 0
Input array:
0 0 0
1 1 1
2 2 2
0 0 0
Output array:
1 1 1
2 2 2
Для продолжения нажмите любую клавишу . . .

```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Enter array's size (n,m)
5 4
Enter array
0 0 0 0
1 1 1 1
2 2 2 2
0 0 0 0
3 3 3 0
Input array:
0 0 0 0
1 1 1 1
2 2 2 2
0 0 0 0
3 3 3 0
Output array:
1 1 1 1
2 2 2 2
3 3 3 0
Для продолжения нажмите любую клавишу . . .
```

Вывод: я научился обрабатывать матрицы и, в частности, удалять из матрицы строки (столбцы)

Декомпозиция и разработка подпрограмм

Цель: научиться декомпозировать и разрабатывать подпрограммы

Задача: определить между какими точками на координатной плоскости самое короткое расстояние и вывести на экран это расстояние.

Структурная схема

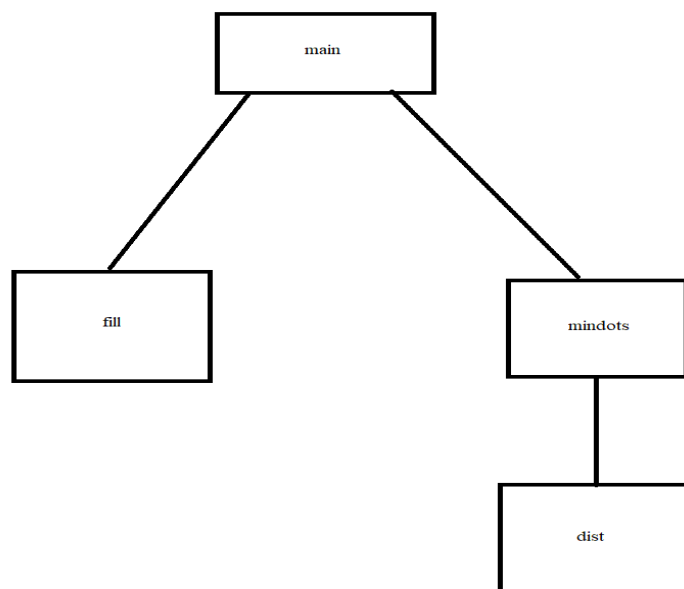
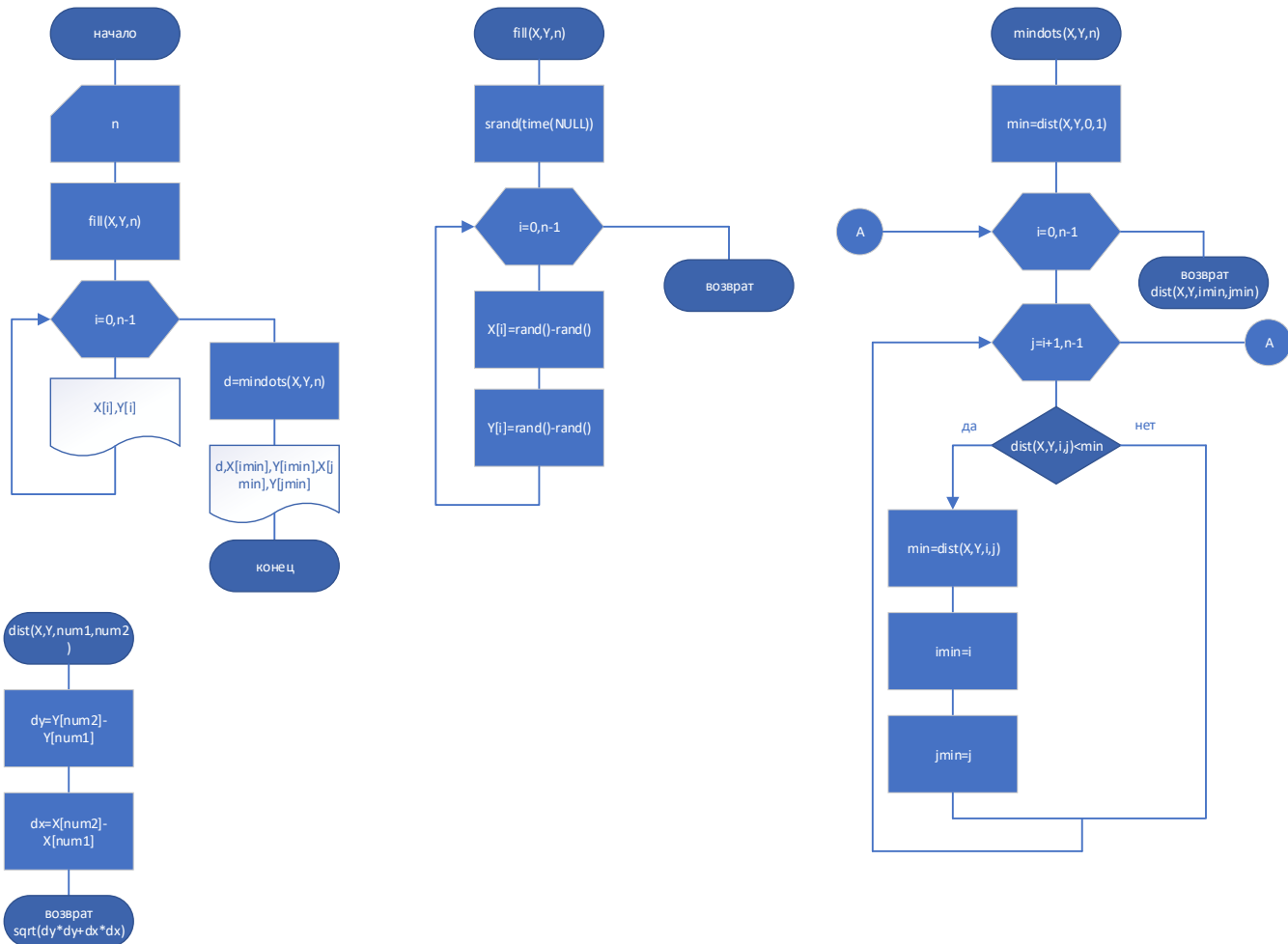


Схема алгоритма



Текст программы

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
int imin,jmin;
void fill(double *X,double *Y,int n) {
    srand(time(NULL));
    for(int i=0;i<n;i++) { X[i]=rand()%51-rand()%51; Y[i]=rand()%51-
    rand()%51; }
}
double dist(double *X,double *Y,int num1,int num2) {
    double dy=Y[num2]-Y[num1];
    double dx=X[num2]-X[num1];
    return sqrt(dy*dy+dx*dx);
}
double mindots(double *X,double *Y,int n) {
    double min=dist(X,Y,0,1);
    for(int i=0;i<n;i++) {
        for(int j=i+1;j<n;j++) {
            if(dist(X,Y,i,j)<min) {
                min=dist(X,Y,i,j);
                imin=i;
                jmin=j;
            }
        }
    }
    return dist(X,Y,imin,jmin);
}
    
```

```

}
void main() {
    int n;
    puts("Enter a number of dots");
    scanf_s("%d", &n);
    while(n<1) {
        puts("Error! A number of dots must be not less than 1");
        scanf_s("%d", &n);
    }
    double *X,*Y;
    X=new double[n];
    Y=new double[n];
    fill(X,Y,n);
    puts("Input dots:");
    for(int i=0;i<n;i++) printf("(%g,%g)\n",X[i],Y[i]);
    double d=mindots(X,Y,n);
    printf("Minimum distance is %g between (%g,%g) and (%g,%g)\n",d,X[imin],Y[imin],X[jmin],Y[jmin]);
    system("pause");
}

```

Тестирование

```

C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Enter a number of dots
25
Input dots:
(5,5)
(-6,12)
(12,14)
(-26,7)
(23,-19)
(3,-30)
(-34,-14)
(-27,-35)
(11,-48)
(17,2)
(9,4)
(4,-16)
(-23,4)
(-9,21)
(-7,-4)
(-9,8)
(6,39)
(-20,9)
(0,-33)
(20,-13)
(-18,37)
(-30,-16)
(6,5)
(12,-17)
(18,22)
Minimum distance is 1 between (5,5) and (6,5)
Для продолжения нажмите любую клавишу . . .

```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Enter a number of dots
20
Input dots:
(3,-18)
(7,-37)
(-4,-12)
(-1,26)
(31,10)
(-25,17)
(36,35)
(-2,-29)
(7,9)
(-34,49)
(3,19)
(-8,5)
(17,-22)
(-2,-17)
(20,10)
(25,22)
(8,16)
(33,19)
(-2,-48)
(23,1)
Minimum distance is 5.09902 between (3,-18) and (-2,-17)
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Enter a number of dots
15
Input dots:
(20,-3)
(-19,-3)
(7,34)
(42,-12)
(-9,-20)
(-20,-6)
(20,-43)
(44,-46)
(35,5)
(35,-25)
(35,-7)
(15,23)
(-6,3)
(-7,1)
(-1,25)
Minimum distance is 2.23607 between (-6,3) and (-7,1)
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Enter a number of dots
10
Input dots:
(-4,7)
(6,11)
(22,-27)
(-9,4)
(-9,-36)
(10,10)
(-11,17)
(8,1)
(2,-31)
(-23,3)
Minimum distance is 4.12311 between (6,11) and (10,10)
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Enter a number of dots
5
Input dots:
(14,-10)
(-6,-3)
(-24,-27)
(0,29)
(19,29)
Minimum distance is 19 between (0,29) and (19,29)
Для продолжения нажмите любую клавишу . . .
```

```
C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Enter a number of dots
0
Error! A number of dots must be not less than 1
50
Input dots:
(40,-26)
(19,12)
(-8,13)
(-1,-4)
(18,13)
(9,13)
(-23,21)
(-3,-11)
(-22,-22)
(16,-1)
(-1,-1)
(23,-20)
(5,-12)
(12,-17)
(5,0)
(-9,-17)
(-13,-18)
(-17,-15)
(5,-17)
(10,4)
(-15,-6)
(14,-30)
(-11,-43)
(18,-32)
(-4,13)
(24,10)
(12,-5)
(22,-4)
(-17,15)
(10,-30)
(-21,-14)
(9,31)
(1,14)
(-21,2)
(-10,9)
(-14,-22)
(-7,-21)
(15,-3)
(16,14)
(-13,-1)
(-10,-6)
(9,33)
(24,-13)
(-5,14)
(-32,2)
(-21,3)
(28,-39)
(-8,-31)
(-9,5)
(-17,13)
Minimum distance is 1 between (-21,2) and (-21,3)
Для продолжения нажмите любую клавишу . . .
```

Вывод: я научился декомпозировать и разрабатывать подпрограммы

Контрольные вопросы

1. Что такое структурные типы данных? (Ответ: структурными типами данных называются такие данные, которые в памяти компьютера определенным образом организованы и имеют средства для работы с такой организованной информацией)

2. Что такое массив и как определяется положение элементов в массиве? (Ответ: массив – это упорядоченная совокупность однотипных данных. Каждому элементу массива соответствует один или несколько индексов, определяющих положение элемента в массиве)

3. Приведите основные методы сортировки одномерных массивов? (Ответ: сортировка с помощью обмена, Шейкерная сортировка)

4. Как в памяти располагаются матрицы? (Ответ: матрицы расположены в памяти “построчно”, т.е. правые индексы меняются быстрее, чем расположенные левее)

5. Какие задачи обработки матриц применимы только для матриц? (Ответ: вычеркивание из матрицы строк (столбцов), удовлетворяющих определенным условиям)

6. Что такое подпрограмма? (Ответ: подпрограмма – это относительно самостоятельный фрагмент алгоритма, соответствующим образом оформленный и снабженный именем)

7. Как подпрограмма получает исходные данные? (Ответ: данные для обработки процедуры и функции получают из вызвавшей их основной программы или подпрограммы)

8. Какие способы передачи одномерных массивов в подпрограммы вы знаете? (Ответ: передача массива и его размера с помощью формальных параметров, передача массива по ссылке)

9. Что такое формальные и фактические параметры подпрограммы? В чем их отличие? (Ответ: формальными называются параметры, определенные в заголовке функции при ее описании. Фактическими называются параметры, задаваемые при вызове функции)

Вывод: я научился обрабатывать одномерные массивы и освоил метод Шейкерной сортировки, научился обрабатывать матрицы, в частности, удалять строки (столбцы) из матрицы, и, наконец, научился декомпозировать и разрабатывать подпрограммы.